On the Usability of Lombardi Graph Drawings

Helen C. Purchase[1], John Hamer[1], Martin Nöllenburg[2,⋆], and Stephen G. Kobourov[3]

[1] School of Computing Science, University of Glasgow, Glasgow, UK
[2] Faculty of Informatics, Karlsruhe Institute of Technology, Karlsruhe, Germany
[3] Department of Computer Science, University of Arizona, Tucson, AZ, USA

Abstract. A recent line of work in graph drawing studies *Lombardi drawings*, i.e., drawings with circular-arc edges and perfect angular resolution at vertices. Little is known about the effects of curved edges versus straight edges in typical graph reading tasks. In this paper we present the first user evaluation that empirically measures the readability of three different layout algorithms (traditional spring embedder and two recent near-Lombardi force-based algorithms) for three different tasks (shortest path, common neighbor, vertex degree). The results indicate that, while users prefer the Lombardi drawings, the performance data do not present such a positive picture.

1 Introduction

Graph drawings with curved edges have seen a renewed interest in recent years, inspired by the work of American abstract artist Mark Lombardi, who created drawings of social networks characterized by the use of curved, almost circular edges and an even distribution of the edges around incident vertices [1]. *Lombardi drawings* are thus defined as graph drawings with perfect angular resolution and circular-arc edges [2]. Previous work on Lombardi drawings includes both theoretical results and practical algorithms for creating *near-Lombardi* drawings, i.e., drawings with circular arcs that aim for good angular resolution.

It remained an open question whether Lombardi drawings, with their smoother and organic shape, just have an aesthetic appeal to humans, who are known to prefer round shapes [3], or whether they actually increase performance in typical graph reading tasks. While readability of graph drawings has been examined before and is known to depend on many factors (experimental studies point to edge crossings and angular resolution as having significant impact [4]), these evaluated only straight-line and polyline drawings. To the best of our knowledge, no empirical studies have been performed on Lombardi drawings.

In this paper we describe the first experimental study comparing traditional straight-line drawings of graphs with near-Lombardi drawings. In Section 2 we sketch the three algorithms used to create our drawings for the visual stimuli. We designed a user study comparing drawings for graphs with different characteristics (size, density, planarity) based on three graph reading tasks (shortest paths, common neighbors, degree estimation); see Section 4. We report and discuss the results obtained from our user experiment in Section 5.

⋆ Supported by the ‚Concept for the Future’ of KIT under project YIG 10-209 within the framework of the German Excellence Initiative.

W. Didimo and M. Patrignani (Eds.): GD 2012, LNCS 7704, pp. 451–462, 2013.

1.1 Related Work

Early work on angular resolution in graph drawing addresses the problem in the straight-line setting [5, 6] or polyline setting [7]. Even with circular-arc edges, maintaining bounded angular resolution leads to exponential area [8]. Force-directed algorithms have been used to maximize the angular resolution using quadratic and cubic Bézier curves [9] and for fixed position drawings with cubic Bézier curves [10]. Curved edges and polyline edges have been studied in the context of drawing planar graphs with good angular resolution [7]. Rotating optimal angular resolution templates for each vertex in the fixed position setting has also been studied [12]. Characterizations and algorithms for special classes of graphs for certain types of Lombardi drawings are known [2, 13], as well as generalizations using circular poly-arcs [14]. Gephi [15] offers a curved edge drawing style, but simply replaces straight edges by curved edges and does not optimize angular resolution.

Recently, two force-directed methods that produce Lombardi (or near-Lombardi) drawings of graphs were proposed by Chernobelskiy et al. [16]. In both cases every edge of the graph is represented by a circular arc and the algorithms aim to maximize the angular resolution at every vertex; implementations of these two algorithms are used in our study. Both algorithms are sketched in Section 2.

2 Overview of the Layout Algorithms

2.1 Straight-line Spring Embedder (TS)

The *traditional spring* algorithm (TS) is an implementation of the well-known Fruchterman-Reingold algorithm [17]. The graph $G = (V, E)$ is modeled as a system of particles, the vertices, on which two kinds of forces act. There is a repulsive force of magnitude $F_r(u, v) = l^2/d(u, v)$ pushing vertex $v \in V$ away from vertex $u \in V$ for every $u \neq v$, where $d(u, v)$ is the current Euclidean distance between u and v and l is the ideal edge length. Moreover, there is an attractive force of magnitude $F_a(u, v) = d(u, v)^2/l$ pulling vertex v toward the adjacent vertex $u \in V$ for every edge $\{u, v\} \in E$. The overall force $F(v)$ acting on a vertex $v \in V$ is the sum of all repulsive and attractive forces. The force computation and subsequent vertex position updates are iterated until the system stabilizes or a specified number of iterations is reached.

2.2 Lombardi Embedder (LE)

This is a straight-forward generalization of the straight-line spring embedder, also referred to as the "tangent-based algorithm" [16]. For each vertex $v \in V$, three appropriately scaled forces are added together with a fourth rotational force in order to determine the overall force $F(v)$ acting on the vertex.

1. A repulsive force $F_r(u, v) = l^2/d(u, v)^3$ pushes vertex $v \in V$ away from vertex $u \in V$ for every $u \neq v$.
2. An attractive force $F_a(u, v) = (d(u, v) - l)/d(u, v)$ pulls vertex v toward the adjacent vertex u for every edge $\{u, v\} \in E$.

3. A tangential force moves vertices connected by edges so as to make circular arc edges possible. Applied to every pair of adjacent vertices, it is defined as $F_t(u, v) = a \times \delta$, where δ is the difference between the current and optimal position of v, and a is a tangential force constant.
4. A rotational force, F_ρ, rotates a vertex and its tangent template so as to make the tangent angles match, making the arc between two vertices possible. It is defined as $F_\rho = b \times \Delta\text{angle}$, where Δangle is the rotation required and b is a rotational constant.

2.3 Restricted Lombardi Embedder (RLE)

This "dummy vertex algorithm" [16], begins with the straight-line drawing obtained from TS. Every edge is subdivided with the addition of a "dummy vertex". Once the endpoints of an edge have been placed and fixed, only one more point is required to uniquely determine a circular arc between these points. Thus, all possible arcs between the vertices can be described by the set of points along the perpendicular bisector of their straight-line connection.

The angular resolution around each vertex is optimized by defining standard repulsive and attractive forces as in the TS method, but using a projection of the force vector to move the dummy vertices along their respective bisectors. Repulsive forces for each dummy vertex are computed with respect to all original vertices and all other dummy vertices. Attractive forces are only defined for the two neighboring vertices of each dummy vertex. The two partial edges incident to each dummy vertex get assigned a fractional resting length as usual. Once the overall force vector $F(v)$ for a dummy vertex v is computed, it is projected on the bisector along which v can move to obtain v's new position. Note that in this second phase only the dummy vertices are moved. It is guaranteed that all edges are circular arcs, and the vertices of G keep the same positions as in the corresponding TS layout.

3 Research Questions

Two primary research questions are addressed in this experiment:

- Do curved edges in a graph drawing assist in the reading of the relational information represented by the graph? This is a comparison between the performance of TS and RLE, which have the same node positioning.
- Does a Lombardi graph drawing layout assist in the reading of the relational information represented by the graph? This is a comparison between the performance of TS against that of LE, where the latter is unconstrained in its node-positioning.

Figure 1 shows the same graph drawn using the three methods. While we have no need to explicitly compare RLE with LE in order to address these research questions, the experiment enables this comparison to be made as well.

In graph drawings with straight lines between nodes, the angles between the edges at each node are fixed once the node positions are set. This may result in very small angles between edges incident at a node, making them hard to distinguish from each

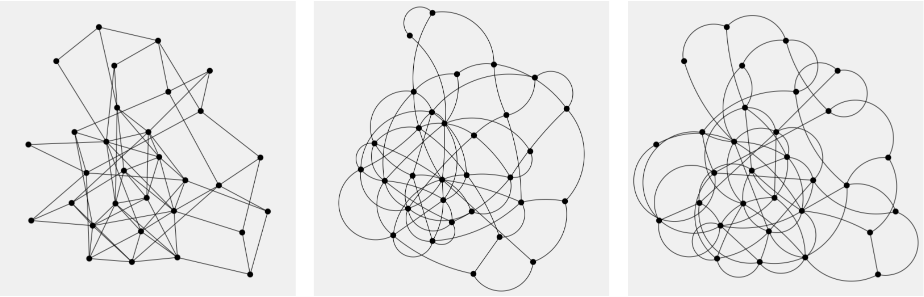


Fig. 1. The same graph drawn with the TS, LE and RLE algorithms

other. The eye can distinguish objects at a minimum acuity of $1'$ [18]; that is, the angle subtended by the eye by the gap between the two objects must be at least $1'$.

Using curved edges means that the angles between edges can be controlled better by the layout algorithm, permitting wider angles between the edges, larger subtending angles, and a better chance for the eye to distinguish them from each other. This would suggest that the use of curved edges would lead to improved interpretation of the relational information, as the relations would be more easily visible.

Our independent variable is therefore the layout of the graph drawing; the three conditions are the spring algorithm (TS), the restricted Lombardi embedder (RLE) and the Lombardi embedder (LE). Prior to the experiment, we had no expectations as to whether the two drawings with curved edges (RLE or LE) would be most beneficial: both use curved edges, and so both should enable edges to be more clearly distinguished at the node than straight-line drawings.

4 Experimental Methodology

4.1 Experimental Objects

In deciding on which graphs to use, we identified three dimensions that have clearly distinguishable values, and which might usefully serve as interesting secondary factors to investigate:

- Graph-theoretic property: **p**lanar or **n**on-planar
- Size: **m**edium (40 nodes) or **s**mall (20 nodes)
- Density: **d**ense (edge:node ratio = 3.5) or **s**parse (edge:node ratio = 1.25)

These dimensions define eight different experimental objects:

	Small		Medium	
	Dense	Sparse	Dense	Sparse
Planar	PSD	PSS	PMD	PMS
Non-planar	NSD	NSS	NMD	NMS

The random graph generation feature of yEd makes it relatively easy to generate graphs that fulfill these parameters[1].

4.2 Experimental tasks

There are a myriad of possible graph reading tasks that could have been used; we chose only three so as to ensure that the whole experiment ran in reasonable time. The criteria used for selecting our tasks were:

- Each task should be different; thus, for example, no two tasks should both require the identification of the shortest path between two nodes.
- Each task should require that participants have to look at individual edges; thus, for example, tasks should not only relate to a global overview of the drawing.
- Each task should be easy to explain to a novice.
- Each task should not be so complicated it would be impossible to answer within a reasonable length of time (maximum of approximately 20 seconds).

We settled on the following three tasks:

The SP task. How long is the shortest path between the two highlighted nodes? A local task that focuses on edges and nodes which tend to lie between the two nodes. The eye will typically move first in a direct line from one node to the other, before widening the search on either side.

The CO task. How many common neighbors do the two highlighted nodes have? Wider than the shortest path task, and less local. The nodes to be counted are less likely to be in the area between the two nodes, and the eye needs to look more at surrounding areas.

The DE task. Which of the three highlighted nodes has the highest degree? A very local task that focuses entirely on the nodes, and their edges; edges are not followed; they are simply counted.

Different versions of the experimental objects were generated for each task/condition combination, and appropriate nodes were chosen for highlighting so as to ensure a wide variety and similar distribution of possible answers for each task. We then laid out each of the 72 graphs according to its associated layout condition, producing the stimuli for the experiment. Figure 4.2 shows some example stimuli.

4.3 Experimental Process

An online experimental system was implemented to present the 72 stimuli and questions for this within-participant experiment, and collect the participants' answers. An experimental session comprised the following process:

- An online consent form.
- A tutorial that introduced graphs, nodes and edges, and which presented worked examples of the three experimental tasks.

[1] yEd is available from `www.yworks.com`

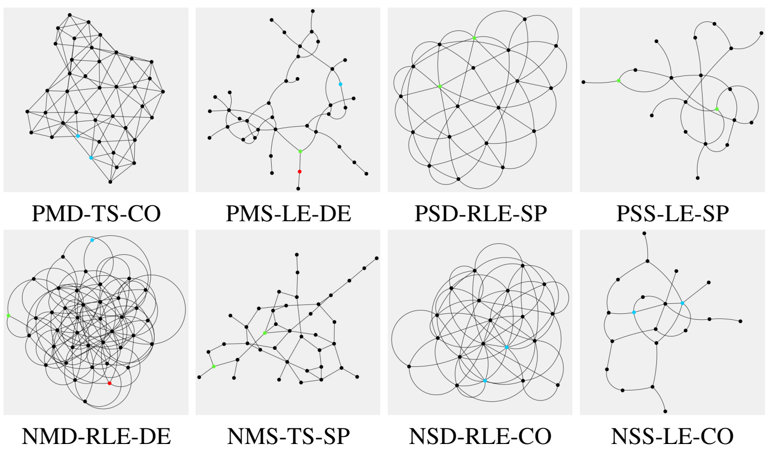
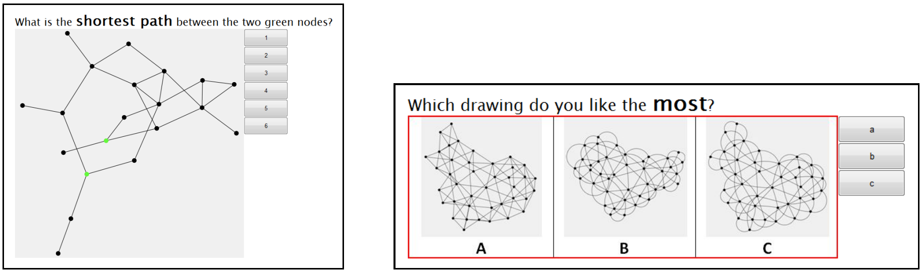


Fig. 2. Example stimuli, labeled by their parameters, condition and task

- Example trials that allowed the participants to practice the three tasks until they got them correct.
- Instructions. Participants were asked to "Please answer the questions as quickly and as accurately as you can."
- Nine practice trials, three for each of the tasks, using drawings similar to the experimental stimuli. The data for these trials were discarded (although the participants did not know this), as their sole purpose was to counter any learning effect.
- The 72 trials, in a different random order for each participant (so as to counter the learning effect), and at a different random orientation (so as to counter any possible orientation confounding factors). There was no time limit. Figure 3 shows two screenshots from the trials.
- After every eight trials, there was a break, when the participant was shown a league table indicating how their overall performance compared with that of other participants. The participants could continue when they were ready.
- Preferences trials, which presented sets of the three different layouts of each of eight experimental objects. Each set was presented twice. In one case, the participant was asked to indicate which drawing they liked the *least*; in the other, the participant indicated which drawing they liked the *most*. The drawings were presented in different random orders and orientation. Figure 3 shows one of these preference trials.
- An invitation for the participant to leave their email address if they wished to be considered for the book token prizes (one for the highest performer, one randomly chosen).

Sixty nine participants took part using the online system over a period of 12 days. Eight participants discussed their preferences with the experimenter in a brief interview after completing the experiment.



What is the shortest path between the two green nodes?

Which drawing do you like the most?

Fig. 3. A typical experimental trial and preference question

5 Results

Each of the 69 participants completed 72 trials: 8 experimental objects × 3 tasks × 3 algorithms. For each trial, the accuracy and response time were recorded, allowing for analysis of two separate performance measures.

5.1 Performance Data

The analysis of the task performance data was conducted using repeated measures Friedman tests with post-hoc pairwise comparisons; the data is not normally distributed, so a non-parametric analysis is appropriate.

The analysis compared the performance of TS, RLE and LE for both time and error data; the results are shown in Table 5.1, with the most important bar charts included in Figure 4. It was not necessary to apply Bonferroni adjustments, as the significance values obtained were almost always < 0.001.

In summary, the significant results are:

- Over all tasks and all graphs: TS produced both more accurate and quicker performance than both LE and RLE. There was no significant performance difference between RLE and LE.
- For the shortest path tasks: TS produced both more accurate and quicker performance than LE, and quicker performance than RLE. The contradictory results between RLE and LE (LE is quicker than RLE and RLE is more accurate than LE) are not due to a time-error trade-off; the correlation between time and error for the SP trials is positive and significant ($r = 0.060, p = 0.015$), indicating that it is not the case that faster responses led to more errors. These two results therefore stand independently.
- For the common neighbour tasks: TS produced both more accurate and quicker performance than LE, and was more accurate than RLE.
- For the degree tasks: RLE produced slower performance than both TS and LE. LE produced better accuracy than both TS and RLE.

When considering the other dimensions (sparse vs. dense, small vs. medium, planar vs. non-planar), in all cases, where significance was found, it indicated that TS performed better than LE and/or RLE (the only exception being the result that RLE was quicker than LE for sparse graphs).

Table 1. Results of the data analysis, showing the means and pairwise significance between each algorithm and dimension for time (in seconds), error rate, and performance (or "n.s." where there is no significance)

		Overall	SP	CO	DE	Sparse	Dense	Pl.	Non-Pl.	Small	Med.
time (sec.)	mean TS	10.94	11.87	13.53	7.40	7.59	14.28	9.35	12.52	9.58	12.29
	mean LE	12.32	13.46	15.61	7.89	9.32	15.32	10.78	13.86	11.86	12.78
	mean RLE	12.38	14.75	14.46	7.93	8.51	16.24	11.63	13.13	10.73	14.03
	LEvTS	<0.001	<0.001	0.003	n.s.	<0.001	0.001	<0.001	n.s.	<0.001	<0.001
	RLEvTS	<0.001	<0.001	n.s.	<0.001	<0.001	0.002	<0.001	n.s.	0.001	n.s.
	RLEvLE	n.s	0.008	n.s.	0.014	0.003	n.s.	n.s.	n.s.	n.s.	n.s.
error rate	mean TS	0.137	0.082	0.190	0.140	0.027	0.248	0.064	0.210	0.167	0.108
	mean LE	0.199	0.239	0.281	0.078	0.082	0.316	0.198	0.200	0.228	0.170
	mean RLE	0.200	0.121	0.310	0.170	0.056	0.345	0.176	0.225	0.200	0.202
	LEvTS	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	n.s.	<0.001	<0.001
	RLEvTS	<0.001	n.s.	<0.001	n.s.	0.002	<0.001	<0.001	n.s.	n.s.	<0.001
	RLEvLE	n.s.	<0.001	n.s.	<0.001	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.
preference	mean TS	0.521				0.534	0.509	0.491	0.552	0.444	0.599
	mean LE	0.560				0.599	0.521	0.657	0.463	0.642	0.478
	mean RLE	0.419				0.368	0.470	0.353	0.485	0.414	0.424
	LEvTS	n.s.				n.s.	n.s.	0.002	n.s.	<0.001	n.s.
	RLEvTS	0.008				<0.001	n.s.	0.011	n.s.	n.s.	0.007
	RLEvLE	<0.001				<0.001	n.s.	<0.001	n.s.	<0.001	n.s.

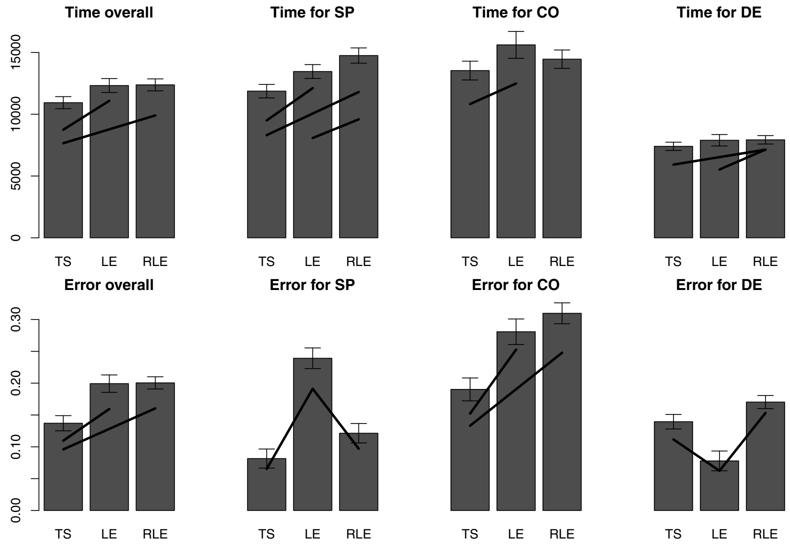


Fig. 4. Bar plots of mean performance for time (in seconds) and error rate. The dark lines show where significant differences were found. The error bars show 95% confidence intervals.

5.2 Preference Data

Each participant chose their most favoured and least favoured drawings. 67 participants completed this stage of the experiment, each producing 16 judgements (8 experimental objects × 2).

We gave a 0 score to each most favoured drawing, 1 for each least favoured drawing, and 0.5 for those drawings for which no judgements were made[2]. This is ordinal data, so a non-parametric analysis is appropriate. As before, Friedman tests were used, comparing the preferences of TS, RLE and LE. The results are shown in Table 5.1.

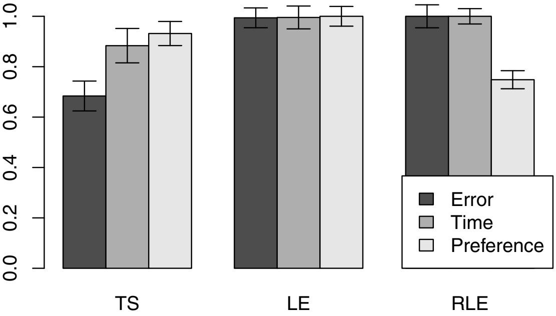


Fig. 5. Relative mean error rate, time and preference for the three algorithms. Values are normalised to lie between 0 and 1 to aid comparison. Smaller means "better".

Overall RLE was significantly preferred to both TS and LE, and this pattern was replicated for both sparse and planar graphs, with no significant preferences indicated for dense and non-planar graphs. When considering the graphs by size, LE was preferred least for small graphs, and RLE preferred more than TS for medium graphs.

There is no significant correlation between the preference judgements and the performance data (time, $r = 0.014$, error, $r = -0.004$). Figure 5 shows the normalised means for time, error and preference. For both time and error, TS produces best performance, while RLE is most preferred.

5.3 Demographic Analysis

At the end of the experiment, participants were asked "How familiar are you with graphs?", with options of "This is my first time", "I have come across them before" and "I am quite familiar with them", and they indicated their gender. Two participants did not provide any demographic data. When considering the two demographic dimensions together, less than 10 percent of the male participants indicated that this was the first time that they had seen graphs, while this proportion was 20 percent for the females.

The differences in performance and preference between TS, LE and RLE data were analysed with respect to these two demographic dimensions. The overall pattern of

[2] This scoring scheme means that high values are associated with 'bad', as with the error and time data.

results for both performance and preference was replicated for male participants ($n = 53$), and for participants who had seen graphs before ($n = 21$), or were quite familiar with them ($n = 39$). There was no significant differences between the three conditions for females ($n = 14$) or participants who had never seen graphs before ($n = 7$).

5.4 Qualitative Data

In post-experiment interviews, eight participants were asked why they liked or disliked graphs drawn with the three algorithms. Several participants said that they had difficulty with determining whether some edges were attached to a node, or simply went through it: this comment applied equally to all algorithms, and was despite the fact that the nodes used in the different task variants had been chosen so that edges relevant to the task did not suffer from this problem. TS was liked for smaller graphs ("good distance between nodes and edges", "easier to find direction of destination with straight lines."), but not for larger graphs ("poor resolution", "many edge overlaps"). For larger graphs, RLE was preferred because "larger angles between edges", "nodes separated nicely", "more spread out", "several paths can be seen", "more pleasant for the eye" but not for smaller ones ("confusing"). LE was seldom preferred: "very wide loops", "central area too dense", "almost collinear nodes." TS was most preferred for SP and CO tasks ("straight lines better to follow over long distances", "neighbours directly visible", "no detours") while LE and RLE were preferred for the DE task ("edges around a node better distributed", "good angular resolution", "can count opposite edges as pairs", although one participant said that "straight edges [were] easier to count then curves".

Other interesting comments left in the online system at the end of the experiment included: "I find it more convenient if the edges are all straight [because it looks like a 3D object in space]", "Finding common neighbors was hard when the edges are curves... when the actual orientation is not the same as the orientation based on edges its confusing", "if feels [as if] my brain was much quicker with the curved graphs", "I prefer straight edges but in dense graphs with long connections rounded edges make sense".

6 Conclusions

These are surprising results indeed. We expected the enhanced angular resolution and visual "flow" of the Lombardi layouts to assist with their interpretation, and for the performance data to reflect this. However, while "tangent based" Lombardi layouts did reduce errors in the degree task, in all other cases the traditional spring layout showed better performance. While we might have expected Lombardi drawings to take longer because of increased edge length[3], the error data which shows that TS produced better accuracy is surprising.

Quantitative data showed that the participants preferred the RLE layout over the straight line alternative, but even the LE layout did not score highly in the preference data. It seems as if the LE layout's use of large arcs around the periphery of the drawing,

[3] An analysis of all drawings indicated a significant difference in the mean number of black pixels between the TS and both RLE and LE, $p = 0.002$.

and the closeness of the central nodes, led to RLE (with its well-spaced nodes and less curvaceous arcs) to be better preferred. In the interviews, participants recognised the benefit of Lombardi over spring drawings for the degree task, while several stated firmly that they preferred straight-line drawings — perhaps as a result of familiarity.

Concurrent experiments on curved graph drawings and Lombardi layouts conducted by Xu et. al [19] also showed no performance difference between Lombardi and straight-line drawings, although, contrary to our results, their participants demonstrated an overwhelming preference for straight-line graphs.

All experiments have limitations, and results must be interpreted within the parameters of this experiment. While we tried to be as diverse as possible with choice of graphs, tasks and participants, it is impossible to cover all options. In addition, participant comments suggested that the scoring mechanism encouraged some participants to answer quickly, deliberately sacrificing accuracy; however, an analysis of all trials indicated no negative correlation between time and errors ($r = 0.181, p < 0.001$), so it is unlikely that time/error trade-off affected the results. Participants taking part in an online experiment in the absence of an experimenter are less likely to take an experiment seriously than those in a scheduled session. However, we found no evidence of this in the data, and believe the scoring mechanism and league tables encouraged valid participation.

Discovering that these Lombardi algorithms do not produce superior performance for tasks requiring edges to be followed does not mean that they have no value. There is much to be said for taking users' preferences into account. HCI studies have shown that if an interface is aesthetically pleasing, users are more willing to persevere in completing a task, even if the interface does not support task performance accuracy or efficiency [20].

User feedback revealed several aspects to be improved in the Lombardi embedders. One possible reason for the better performance of straight-line drawings for path-finding tasks is that straight edges do not detour, and hence have a narrower visual search range for paths. To counteract this, one could limit the curvature of edges and reduce the required search range, compromising on the angular resolution while remaining aesthetically pleasant. Particularly, it would be interesting to cater to the remark expressed by participants that they liked Lombardi drawings with sequences of curved edges forming smooth visual paths. This might lower the visual complexity compared to polygonal curves with sharp bends. Explicitly avoiding unrelated nodes and edges that come too close to each other is a further point for improvement, one that fits well into the force-based framework.

In addition, this experiment has only looked at the effect of curved edges. Future work could consider also the effect of the combination of curved edges and other layout aesthetics: for example, is it better to have a curved edge drawing with few edge crossings than a straight line one with many? It might be that any performance gain from the use of straight line drawings is countered by other visual features of a spring layout, and that the enhanced motivation arising from using a preferred layout means that Lombardi algorithms are the better choice after all.

Despite these strong results, there is still a place for Lombardi graph drawings, in situations where users have time to focus on accuracy and to enjoy the elegant aesthetic.

Acknowledgments. We are grateful to Roman Chernobelskiy for providing and adapting the algorithms for our experiment, and to the participants of this study.

References

1. Hobbs, R.: Mark Lombardi: Global Networks. Independent Curators (2003)
2. Duncan, C.A., Eppstein, D., Goodrich, M.T., Kobourov, S.G., Nöllenburg, M.: Lombardi drawings of graphs. J. Graph Alg. and Applications 16(1), 85–108 (2012)
3. Bar, M., Neta, M.: Humans prefer curved visual objects. Psychological Science 17(8), 645–648 (2006)
4. Purchase, H.: Which Aesthetic Has the Greatest Effect on Human Understanding? In: DiBattista, G. (ed.) GD 1997. LNCS, vol. 1353, pp. 248–261. Springer, Heidelberg (1997)
5. Di Battista, G., Vismara, L.: Angles of planar triangular graphs. SIAM J. Discrete Math. 9(3), 349–359 (1996)
6. Garg, A., Tamassia, R.: Planar Drawings and Angular Resolution: Algorithms and Bounds. In: van Leeuwen, J. (ed.) ESA 1994. LNCS, vol. 855, pp. 12–23. Springer, Heidelberg (1994)
7. Gutwenger, C., Mutzel, P.: Planar Polyline Drawings with Good Angular Resolution. In: Whitesides, S.H. (ed.) GD 1998. LNCS, vol. 1547, pp. 167–182. Springer, Heidelberg (1999)
8. Cheng, C., Duncan., C., Goodrich, M., Kobourov, S.: Drawing planar graphs with circular arcs. Discrete Comput. Geom. 25(3), 405–418 (2001)
9. Finkel, B., Tamassia, R.: Curvilinear Graph Drawing Using the Force-Directed Method. In: Pach, J. (ed.) GD 2004. LNCS, vol. 3383, pp. 448–453. Springer, Heidelberg (2005)
10. Brandes, U., Schlieper, B.: Angle and Distance Constraints on Tree Drawings. In: Kaufmann, M., Wagner, D. (eds.) GD 2006. LNCS, vol. 4372, pp. 54–65. Springer, Heidelberg (2007)
11. Gutwenger, C., Mutzel, P.: Planar Polyline Drawings with Good Angular Resolution. In: Whitesides, S.H. (ed.) GD 1998. LNCS, vol. 1547, pp. 167–182. Springer, Heidelberg (1999)
12. Brandes, U., Shubina, G., Tamassia, R.: Improving angular resolution in visualizations of geographic networks. In: Proc. 2nd TCVG Symp. Visualization (VisSym 2000), Springer, 23–32 (2000)
13. Duncan, C., Eppstein, D., Goodrich, M., Kobourov, S., Nöllenburg, M.: Drawing Trees with Perfect Angular Resolution and Polynomial Area. In: Brandes, U., Cornelsen, S. (eds.) GD 2010. LNCS, vol. 6502, pp. 183–194. Springer, Heidelberg (2011)
14. Duncan, C.A., Eppstein, D., Goodrich, M.T., Kobourov, S.G., Löffler, M.: Planar and poly-arc lombardi drawings. In: Speckmann, B. (ed.) GD 2011. LNCS, vol. 7034, pp. 308–319. Springer, Heidelberg (2011)
15. Bastian, M., Heymann, S., Jacomy, M.: Gephi: An open source software for exploring and manipulating networks. In: Proc. 3rd Int. Conf. Weblogs and Social Media, pp. 361–362 (2009)
16. Chernobelskiy, R., Cunningham, K.I., Goodrich, M.T., Kobourov, S.G., Trott, L.: Force-Directed Lombardi-Style Graph Drawing. In: Speckmann, B. (ed.) GD 2011. LNCS, vol. 7034, pp. 320–331. Springer, Heidelberg (2011)
17. Fruchterman, T., Reingold, E.: Graph drawing by force-directed placement. Software Practice and Experience 21(11), 1129–1164 (1991)
18. Ware, C.: Info. Visualization: Perception for Design. Morgan Kaufmann (2004)
19. Xu, K., Rooney, C., Passmore, P., Ham, D.H., Nguyen, P.: A user study on curved edges in graph visualization. In: IEEE InfoVis (2012) (to appear)
20. Nakarada-Kordic, I., Lobb, B.: Effect of perceived attractiveness of web interface design on visual search of web sites. In: Proc. 6th SIGCHI NZ Conf. Computer-Human Interaction (CHINZ 2005), pp. 25–27. ACM, New York (2005)